

Automate all the things!

Abenteuer auf dem Weg zu Continuous Deployment mit Sitecore

Agenda

- Ein Überblick
 - Was nicht funktioniert
 - Zweiter Versuch: Grundlagen
 - Herausforderungen im Kontext Sitecore
-

Ein Überblick

Maturlitätsmodelle zu Continuous Deployment

	Base	Beginner	Intermediate	Advanced	Expert
Culture & Organization	<ul style="list-style-type: none"> Prioritized work Defined and documented process Frequent commits 	<ul style="list-style-type: none"> One backlog per team Share the pain Stable teams Adopt basic Agile methods <div> <ul style="list-style-type: none"> Library Management Version Control DB Changes </div>	<ul style="list-style-type: none"> Extended team collaboration Component ownership Act on metrics Remove boundary dev & ops Common process for all changes Centralize decisions 	<ul style="list-style-type: none"> Dedicated tools team Team responsible all the way to prod Deploy disconnected from Release Continuous improvement (Kaizen) 	<ul style="list-style-type: none"> Cross functional teams No rollbacks (always roll forward)
Design & Architecture	<ul style="list-style-type: none"> Consolidated platform & technology 	<ul style="list-style-type: none"> Organize system into modules API management Library management Version control DB changes 	<ul style="list-style-type: none"> No (or minimal) branching Branch by abstraction Configuration as code Feature hiding Making compon <div> <ul style="list-style-type: none"> Automated Tag & Versioning Auto triggered builds </div>	<ul style="list-style-type: none"> Full component ba Push business metrics <div> <ul style="list-style-type: none"> Dedicated tools team </div>	
Build & Deploy	<ul style="list-style-type: none"> Versioned code base Scripted builds Basic scheduled builds (CI) Dedicated build server Documented manual deploy Some deployment scripts exists 	<ul style="list-style-type: none"> Polling builds Builds are stored Manual tag & versioning First step towards standardized deploys 	<ul style="list-style-type: none"> Auto triggered build (commit hooks) Automated tag & versioning Build once deploy anywhere Automated bulk of DB changes Basic pipeline with deploy to prod Scripted config changes (server) process for all environments <div> <ul style="list-style-type: none"> Automatic Integration Tests </div>	<ul style="list-style-type: none"> Zero downtime deploys Multiple build machines Full automatic DB deploys 	<ul style="list-style-type: none"> Build bakery Zero touch continuous deployments <div> <ul style="list-style-type: none"> Zero touch continuous deployments </div>
Test & Verification	<ul style="list-style-type: none"> Automatic unit tests Separate test environment 	<ul style="list-style-type: none"> Automatic integration tests 	<ul style="list-style-type: none"> Automatic component tests (isolated) Some automatic acceptance tests 	<ul style="list-style-type: none"> Full automatic acceptance tests Automatic performance tests Automatic security tests Risk based manual testing 	<ul style="list-style-type: none"> Verify expected business value
Information & Reporting	<ul style="list-style-type: none"> Baseline process metrics Manual reporting 	<ul style="list-style-type: none"> Measure the process Static code analysis Scheduled quality reports 	<ul style="list-style-type: none"> Common information model Traceability built into pipeline Report history is available 	<ul style="list-style-type: none"> Graphing as a service Dynamic test coverage analysis Report trend analysis 	<ul style="list-style-type: none"> Dynamic graphing and dashboards Cross silo analysis

«Start Small & Continuously Improve»

Was nicht funktioniert

Neuland, Glatteis und Lessons Learned



Automatisieren ohne Erfahrung im Prozess

- Manuelle Abläufe, die nur für das Deployment notwendig sind,
 - gehen vergessen,
 - sind zu wenig getestet,
 - brauchen viel Zeit.
 - Unerprobte Abläufe
 - sind aufwändig umzusetzen (fehlendes Know-how),
 - bedeuten ständige Anpassungen (oder schränken ein),
 - machen die Automatisierung komplex & teuer in der Wartung.
-

Die falsche Technologie einsetzen

- MSBUILD
 - Schlechte Entwicklungsumgebung
 - Schwer zu debuggen
 - Schwerfällig
 - Jenkins
 - Man verzichtet auf zu viele Funktionen
 - Monolithische Architekturen
 - Schaffen von ungewünschten Abhängigkeiten innerhalb des Tools
 - Schwerfällig bei Änderungen
-

*„Wer als Werkzeug nur einen Hammer hat,
sieht in jedem Problem einen Nagel“*

Paul Watzlawick

Lessons Learned

- Standard Software Features nutzen
 - Build Steps von Build Server nutzen, nicht mehrere Steps in einem «scripten»
 - VS Publish nutzen, nicht mit selbstgebaudem (MSBUILD) Script Files kopieren
 - Kleine modulare Komponenten als Bindeglieder im Prozess
 - «günstig» ersetzbar
 - schützen vor gefährlichen Abhängigkeiten im eigenen Werkzeugkasten
 - Alles kann automatisiert werden, wenn man ein klares Bild vom Prozess hat.
-

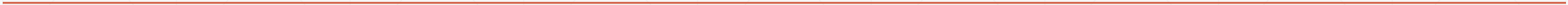
2. Versuch: Grundlagen

Erfahrung ist ein wichtiger Erfolgsfaktor für gelungene Automatisierung

Konzepte und Methoden

- Gitflow
 - Flexibel und dennoch mit klaren Regeln
 - Semantic Versioning (SemVer)
 - konsistente Versionierung
 - dokumentiert sich in wesentlichen Teilen selbst («semantisch»)
 - Passt bestens zum Gitflow Branching Modell
 - Sitecore
 - Solution «ausserhalb des Webroots»
 - Nur was wir erstellen, gehört in die Versionskontrolle
 - Item Serialisierung: VCS ist führend für Änderungen an Items
-

Zentrale Software & Technologie





Pull request

#3

MERGED

update

→

master

Updated/added gitignore files

1 Reviewer

Overview

Diff

Commits

Details

Unwatch this pull request

Learn more

Reto Hugi

created a pull request

11 Aug 2015

- updated .gitignore with latest from gitignore.io
- add default gitignore for serialization folder

Activity

What do you want to say?

Ernst Joss

MERGED

update

to

master

in commit 0716923b519

11 Aug 2015

Ernst Joss

APPROVED

the pull request

11 Aug 2015

Reto Hugi

Hi Aschi, glad you asked :)

That's why: <https://github.com/github/gitignore/commit/68f746b320a8732a7f49d4f4563dcec22596bb4f>

and that's where: <https://github.com/github/gitignore/commit/72e24c5ba6587ddeb89ae5de3b4098ff4bc50c91>

Reply · Edit · Delete · Create task · 11 Aug 2015

Ernst Joss

commented on a file

11 Aug 2015

.gitignore

15	18	[Dd]ebugPublic/
16	19	[Rr]elease/
17	20	[Rr]eleases/
18	21	x64/
19	22	x86/
20	23	build/
21	24	bld/
22	25	[Bb]in/

Stash

Pull Requests / Reviews

Berechtigungen (Projekt, Repo, Branch)

Tc

Projects

Changes

Agents1

Build Queue0

Reto Hugi

Administration

Hide Successful Configurations

Configure Visible Projects

Technology

Root project for technology specific framework and utility development

no hidden

Sitecore

Parent Node for all Sitecore related Projects, frameworks and utility development efforts.

no hidden

Frameworks

no hidden

Sitecore Modules

1 failing31 successful

no hidden

Bob Open Build

no hidden

Tools

2 errors2 successful

no hidden

Rubble

1 successful

no hidden

Packer

Build Sitecore NuGet packages

<Active branches>

no hidden

Build

Run

pull-requests/32

#1.0.1-PullRequest.6

Success

Artifacts

Ernst Joss (2)

one day ago (3m:38s)

SCMBOB-548

#1.0.1-SCMBOB-548.5

Success

Artifacts

Ernst Joss (1)

one day ago (54s)

master

#1.0.1+4

Success

Artifacts

Changes (2)

11 days ago (1m:05s)

Sitecore Release

Pending (2)

Run

master

#156

Success, Sitecore 8.1 revision 151003

Artifacts

Changes (2)

11 days ago (19m:27s)

Sitecore Support

Pending (2)

Run

<default>

#109

Success

Artifacts

Ernst Joss (2)

2 days ago (1m:33s)

Third party libraries

No builds to display

Pending (58)

Idle

Run

Publish Third party libraries

Pending (6)

Run

master

#53

Success, Artifact Unic.UrlMapper.6.4.1.3.nupkg

Artifacts

No changes

one month ago (4s)

Benny

1 successful

no hidden

Dodger

1 successful

no hidden

FrontendTest

<Active branches>

2 errors5 successful

no hidden

Config

<Active branches>

1 successful

no hidden

Muck

1 successful

no hidden

Lofty

1 successful

no hidden

TcLegacy

1 successful

no hidden

Scoop

<Active branches>

1 successful

no hidden

TeamCity


Build Server

Amazon EC2 Build Agents

Vererbare Build Konfigurationen

NuGet Feed (inkl. Access Control)

14 Build Steps für «Build» Template

 Octopus Deploy

DashboardEnvironmentsProjectsLibraryTasksadminConfiguration

VibesProcessCreate release

OverviewProcessVariablesReleasesSettings

Deployment process

In order to deploy your applications and services, you need to define your *deployment process*. You can think of your deployment process as being a bit like a recipe, which Octopus will follow when deploying your software. Your deployment process is made up of one or more *steps*. Octopus runs each step in sequence, so it's important to put them in the right order.

1. Lofty

Deploy NuGet package **Unic.Bob.Lofty** from **Teamcity** to machines in roles: **sitecore** **offline**

2. Deploy

Rolling deployment across machines in roles: **sitecore** **offline**

2.1. Backup Webroot

Run a PowerShell script

Only in: Vibes Test Author Vibes Test Delivery Vibes Prod Author Vibes Prod Delivery

2.2. Cleanup Backup folder

Run a PowerShell script

Only in: Vibes Test Author Vibes Test Delivery Vibes Prod Author Vibes Prod Delivery

2.3. Stop IIS AppPool

Run a PowerShell script

Only in: Vibes Test Author Vibes Test Delivery Vibes Prod Author Vibes Prod Delivery

2.4. Backup unmanaged files

Run a PowerShell script

Only in: Vibes Test Author Vibes Test Delivery Vibes Prod Author Vibes Prod Delivery

2.5. Install Sitecore

Deploy NuGet package **Sitecore.Distribution.#{Type}** from **Teamcity**

2.6. Deploy Website

Deploy NuGet package **#{Packageld}** from **Teamcity**

2.7. Restore unmanaged files

Run a PowerShell script

Lifecycle?

Lifecycles define how this project can be promoted between environments. Lifecycles can be defined in the [Library](#).

Vibes ([choose lifecycle](#))

Test

Vibes Test Author

Vibes Test Delivery

Prod

Vibes Prod Author

Vibes Prod Delivery

[View lifecycle definition](#)

Automatic Release Creation

Automatic release creation defines if when a package used in this project is pushed to the built-in NuGet repository a new release should be created.

No deployment step that uses the built-in NuGet repository has been added.

Script modules

The Library includes *PowerShell* modules that can be shared between projects. View, create and modify the contents of variable sets in the [Library](#).

[Include script modules](#)

Octopus Deploy 2.6.3.886DownloadsDocumentationHelp & Support




























OctopusDeploy

Deployment Server

Online & Offline Deployments

Nightlies von TC getriggert

15 Deployment Steps

- +  RadEditor.Net2.8.1.151003.156.nupkg (372.79 KB)
- +  Sitecore.8.1.151003.156.nupkg (3.99 KB)
- +  Sitecore.Abstractions.8.1.151003.156.nupkg (15.37 KB)
- +  Sitecore.Analytics.8.1.151003.156.nupkg (341.78 KB)
- +  Sitecore.Analytics.Core.8.1.151003.156.nupkg (25.66 KB)
- +  Sitecore.Analytics.Model.8.1.151003.156.nupkg (46.39 KB)
- +  Sitecore.Buckets.8.1.151003.156.nupkg (141.26 KB)
- +  Sitecore.Client.8.1.151003.156.nupkg (573.87 KB)
- +  Sitecore.ContentSearch.8.1.151003.156.nupkg (210.25 KB)
- +  Sitecore.ContentSearch.Linq.8.1.151003.156.nupkg (59.52 KB)
- +  Sitecore.Databases.8.1.151003.156.nupkg (58.12 MB)
- +  Sitecore.Distribution.Mvc.8.1.151003.156.nupkg (203.88 MB)
- +  Sitecore.ExperienceEditor.8.1.151003.156.nupkg (97.87 KB)
- +  Sitecore.HtmlAgilityPack.8.1.151003.156.nupkg (2.49 KB)
- +  Sitecore.ItemWebApi.8.1.151003.156.nupkg (28.1 KB)
- +  Sitecore.Kernel.8.1.151003.156.nupkg (1.61 MB)
- +  Sitecore.Logging.8.1.151003.156.nupkg (73.08 KB)
- +  Sitecore.Lucene.Net.8.1.151003.156.nupkg (304.11 KB)
- +  Sitecore.Mvc.8.1.151003.156.nupkg (72.89 KB)
- +  Sitecore.Mvc.Analytics.8.1.151003.156.nupkg (13.53 KB)
- +  Sitecore.Mvc.Config.8.1.151003.156.nupkg (7.84 KB)
- +  Sitecore.Newtonsoft.JSON.8.1.151003.156.nupkg (2.49 KB)
- +  Sitecore.Nexus.8.1.151003.156.nupkg (50.18 KB)
- +  Sitecore.Security.AntiCsrf.8.1.151003.156.nupkg (16.42 KB)
- +  Sitecore.Speak.Charting.8.1.151003.156.nupkg (13.03 KB)
- +  Sitecore.Speak.Client.8.1.151003.156.nupkg (114.73 KB)
- +  Sitecore.Speak.ItemWebApi.8.1.151003.156.nupkg (26.14 KB)

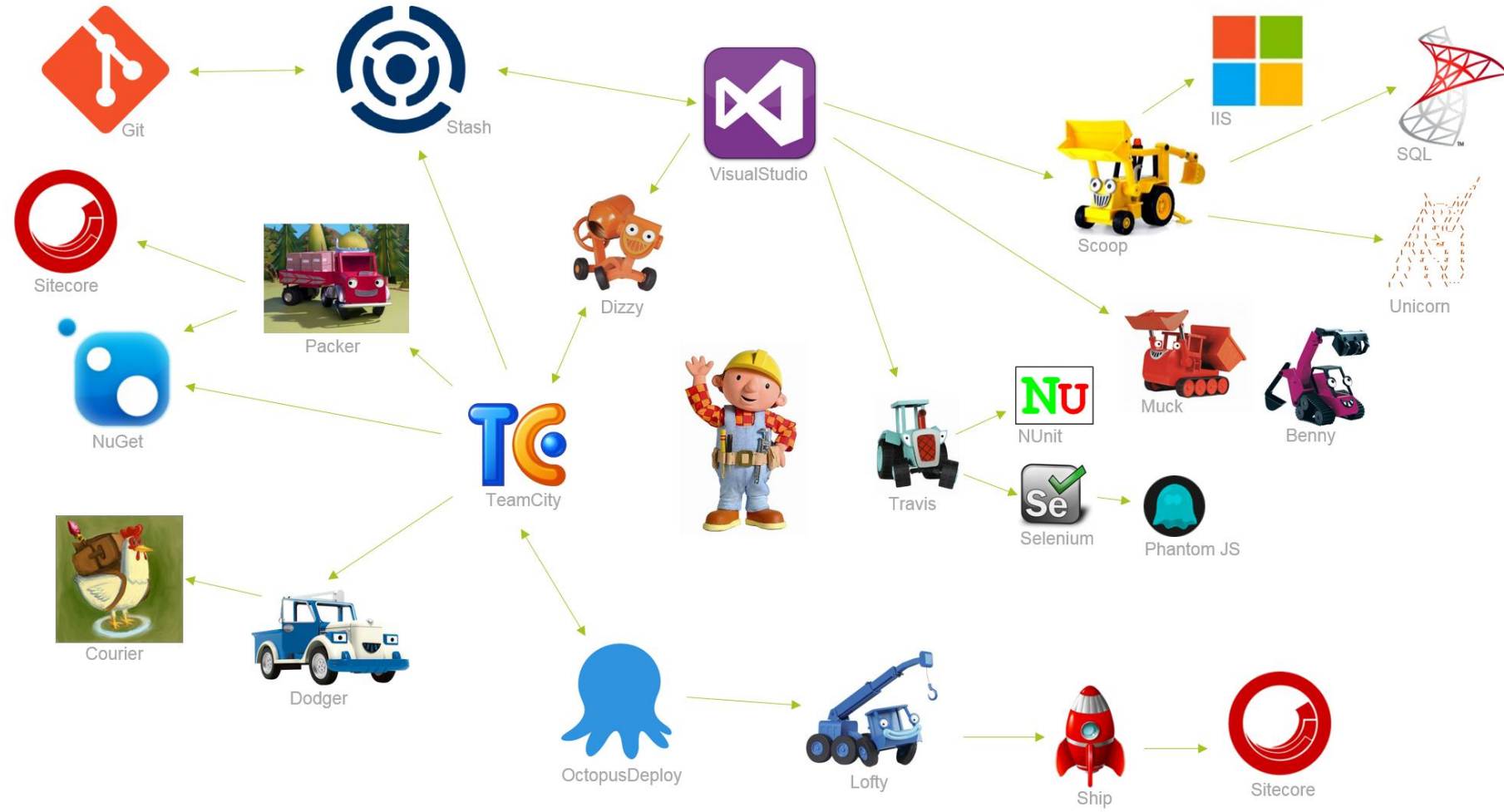
NuGet

Dependency Management

Versionierte Artefakte

Artefakt-Transport

Der Rest: Powershell und Tools




Herausforderungen & Lösungen

Eine Auswahl von Continuous Deployment Hürden

Configuration Management

- «include»-Configs
- Zwei Rollen (Management & Delivery)
- N Umgebungen (Dev, Preview, Test, Integration, Produktion usw.)



10_base
11_local
12_dev
13_igr
14_prod
20_author
20_delivery
30_dev.author
30_dev.delivery
40_igr.author
40_igr.delivery
50_prod.author
50_prod.delivery
90_author
90_delivery
90_local



```
<GlobalWebPath>d:\web</GlobalWebPath>  
<KeepAppConfigIncludes>  
  *_sitecore**;  
  *_base;  
  *_wffm;  
  *_hardening;  
  *_$Environment;  
  *_$Role;  
  *_$Environment.$Role  
</KeepAppConfigIncludes>  
<BennyFileWatcher>  
  *.cshtml;  
</BennyFileWatcher>
```

Versionierung

- Ein Release manuell versionieren scheitert
- Ziel:
 - Automatisch (CI Server)
 - Gemäss SemVer
 - Im Kontext von Gitflow
 - Kompatibel mit NuGet



GitVersion



master	#1.6.0
release/1.6	#1.6.0-beta.1+0
develop	#1.6.0-unstable.26+26
develop	#1.6.0-unstable.25+25
develop	#1.6.0-unstable.24+24
develop	#1.6.0-unstable.23+23

Sitecore Item Management

- Serialization mit Unicorn 2
 - Serialization/app
 - Serialization/appDefault
 - Serialization/testData
 - Sitecore Courier
 - Erstellt item.update packages (inkrementel)
 - TeamCity Integration, Vergleich mit letztem Release
 - Sitecore Ship
 - Importiert item.update packages, die via OctopusDeploy, cURL & Powershell Script installiert werden
-

Sitecore Item Management

- Serialization mit Version 2
 - Serialization
 - Serialization/appData
 - Serialization/testData
- Sitecore Courier
 - Erstellt item.update packages
 - TeamCity Integration mit System Release
- Sitecore Ship
 - Importiert item.update packages, die via OctopusDeploy, cURL oder Powershell Script installiert werden

Sitecore Item Management

- Serialization mit Unicorn 3
 - Serialization/app
 - Serialization/appDefault
 - Serialization/testData
 - TeamCity
 - NuGet Package der Website enthält auch die serialisierten Items aus «app» und «appDefault»
 - Unicorn Sync via OctopusDeploy & PowerShell Script auf Management Server
-

Merci!

Reto Hugi

Sitecore MVP 2013 - 2015

twitter.com/retohugi
github.com/retohugi

Was sonst noch geschah

Kleine Highlights

- Automatisierte «Installation» und Konfiguration einer Sitecore Lösung
 - Vereinfachtes VS Publishing & Filewatcher
 - Automatische Integrationstests
 - Trennung von Frontend und Backend
 - «Install-Frontend»
 - Installer für Offline Deployments
-